



AFRL-VA-WP-TP-2007-317

NEURAL DYNAMIC TRAJECTORY DESIGN FOR REENTRY VEHICLES (PREPRINT)

Ajay Verma, Peng Xu, Kalyan Vadakkevedu, and Rick Mayer
Knowledge Based Systems, Inc.

JULY 2007

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
AIR VEHICLES DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-VA-WP-TP-2007-317 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

Michael W. Oppenheimer
Electronics Engineer
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

//Signature//

Deborah S. Grismer
Chief
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show “//Signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YY) July 2007		2. REPORT TYPE Conference Paper Preprint		3. DATES COVERED (From - To) 15 June 2005 – 23 July 2007		
4. TITLE AND SUBTITLE NEURAL DYNAMIC TRAJECTORY DESIGN FOR REENTRY VEHICLES (PREPRINT)				5a. CONTRACT NUMBER FA8650-06-C-3505		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER 0605502		
6. AUTHOR(S) Ajay Verma, Peng Xu, Kalyan Vadakkevedu, and Rick Mayer (Knowledge Based Systems, Inc.) Michael W. Oppenheimer and David B. Doman (AFRL/VACA)				5d. PROJECT NUMBER A05G		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER 0C		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Knowledge Based Systems, Inc. Control Design and Analysis Branch (AFRL/VACA) College Station, TX 77840 Control Sciences Division Air Force Research Laboratory, Air Vehicles Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force				8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Air Vehicles Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/VACA		
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-VA-WP-TP-2007-317		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.						
13. SUPPLEMENTARY NOTES Conference paper submitted to AIAA GNC Conference proceedings. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. PAO Case Number: AFRL/WS 07-1773, 03 Aug 2007. Paper contains color.						
14. ABSTRACT The next generation of reentry vehicles is envisioned to have onboard autonomous capability of real-time trajectory planning to provide capability of responsive launch and delivering payload anywhere with precise flight termination. This capability is also desired to overcome, if possible, in-flight vehicle damage or control effector failure resulting in degraded vehicle performance. An aerial vehicle is modeled as a nonlinear multi-input-multi-output (MIMO) system. An ideal optimal trajectory control design system generates a series of control commands to achieve a desired trajectory under various disturbances and vehicle model uncertainties including aerodynamic perturbations caused by geometric damage to the vehicle. Conventional approaches suffer from the nonlinearity of the MIMO system, and the high-dimensionality of the system state space. In this paper, we apply a Neural Dynamic Optimization (NDO) based approach to overcome these difficulties. The core of an NDO model is a multilayer perceptron (MLP) neural network, which generates the control parameters online. The advantage of the NDO system is that it is very fast and gives the trajectory almost instantaneously. The bulk of the time consuming computation is required only during off-line training. The inputs of the MLP are the time-variant states of the MIMO systems. The outputs of the MLP are the near optimal control parameters.						
15. SUBJECT TERMS Trajectory Reshaping, Trajectory Planning, Reusable Launch Vehicles						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 18	19a. NAME OF RESPONSIBLE PERSON (Monitor) Michael W. Oppenheimer 19b. TELEPHONE NUMBER (Include Area Code) N/A	
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified				

Neural Dynamic Trajectory Design for Reentry Vehicles

Ajay Verma¹, Peng Xu², Kalyan Vadakkeveedu³, Rick Mayer⁴
Knowledge Based Systems, Inc., College Station, TX, 77840

Michael W. Oppenheimer⁵ and David B. Doman⁶
US Air Force, WPAFB, OH 45433-7531

The next generation of reentry vehicles is envisioned to have onboard autonomous capability of real-time trajectory planning to provide capability of responsive launch and delivering payload anywhere with precise flight termination. This capability is also desired to overcome, if possible, in-flight vehicle damage or control effector failure resulting in degraded vehicle performance. An aerial vehicle is modeled as a nonlinear multi-input-multi-output (MIMO) system. An ideal optimal trajectory control design system generates a series of control commands to achieve a desired trajectory under various disturbances and vehicle model uncertainties including aerodynamic perturbations caused by geometric damage to the vehicle. Conventional approaches suffer from the nonlinearity of the MIMO system, and the high-dimensionality of the system state space. In this paper, we apply a Neural Dynamic Optimization (NDO) based approach to overcome these difficulties. The core of an NDO model is a multilayer perceptron (MLP) neural network, which generates the control parameters online. The advantage of the NDO system is that it is very fast and gives the trajectory almost instantaneously. The bulk of the time consuming computation is required only during off-line training. The inputs of the MLP are the time-variant states of the MIMO systems. The outputs of the MLP are the near optimal control parameters.

I. Introduction

THE next generation of reentry vehicles is envisioned to have onboard autonomous capability of real-time trajectory planning to provide capability of responsive launch and delivering payload anywhere with precise flight termination. This capability is also desired to overcome, if possible, in-flight vehicle damage or control effector failure resulting in degraded vehicle performance. Determining real-time trajectory for highly non-linear reentry vehicles, such as reusable launch vehicles (RLVs), and trajectory guidance has been of considerable research interest⁵ in the recent past. Conventional approach for mission operation for highly non-linear systems consists of offline reference trajectory design and onboard tracking of the reference trajectory. However, in case the vehicle's dynamic behavior is altered significantly due to damage/failure to the vehicle or vehicle sub-system, a pre-planned trajectory may cease to be feasible, possibly resulting in a catastrophic failure and loss of vehicle. The traditional trajectory design approach must be augmented to provide real-time trajectory redesign capability particularly for fully autonomous vehicles. With this motivation we investigated the performance of a neural net based approach for real-time trajectory design

A feasible descent trajectory must lie within an entry corridor defined by the path constraints based on acceptable limits of thermal, structural and operational constraints. The trajectory design system determines the necessary commands required to maneuver the vehicle on a feasible trajectory to accomplish a desired objective. More specifically, given an initial state of the vehicle, the trajectory design system generates a history of control

¹ Research Scientist, 1408 University Dr, College Station, TX-77840, Senior Member AIAA, averma@KBSI.com.

² Researcher, 1408 University Dr, College Station, TX-77840, Member AIAA, pxu@KBSI.com.

³ Researcher, 1408 University Dr, College Station, TX-77840, Member AIAA, kvadakkeveedu@KBSI.com.

⁴ Senior Research Scientist, 1408 University Dr, College Station, TX-77840, rmayer@KBSI.com,

⁵ Electronics Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, WPAFB, OH 45433-7531, Senior Member AIAA, Michael.Oppenheimer@wpafb.af.mil

⁶ Senior Aerospace Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, WPAFB, OH 45433-7531, Associate Fellow AIAA, David.Doman@wpafb.af.mil

inputs so that the vehicle can reach the desired final boundary conditions while imposing all the trajectory constraints so that the designed vehicle trajectory is feasible.

The goal of this paper is to generate an approximate optimal trajectory using a Neural Dynamic Optimization^{1,2,3} (NDO) based approach that provides an approximate technique to solve the *dynamic programming* (DP) problem for multi-input-multi-output (MIMO) discrete systems. Dynamic programming finds an optimal feedback solution for the trajectory of a non-linear MIMO system. However, DP solution can be very difficult for higher order systems and impractical for real time applications. In NDO approach, a more practical method is applied to handle the complexities of the non-linearity of MIMO system. It leverages the advantage of neural networks in the framework of optimal control theory for a feed back solution of the MIMO control systems.

NDO shows distinct advantages compared to other optimal control approaches such as Feedforward Optimal Control (FOC) and Dynamic Programming (DP). FOC is capable of finding an optimal control solution relatively easily. Its computational load is tractable. However, FOC finds just a single trajectory for a given initial state. Thus, its solution is very sensitive to disturbances and cannot handle uncertainties. DP is capable of providing an optimal solution that can handle disturbances and uncertainties. However, real-time computation and storage requirements associated with DP solutions can be challenging, especially for high-order nonlinear systems. NDO overcomes both problems by using a neural network, which is more robust to uncertainties, as the controller and provides real-time numerical computation capability. The design of NDO is similar to FOC; meaning that it is computationally tractable. The difference is that a FOC approach generates the trajectory directly based on the initial state, while an NDO trains a neural network whose responsibility is to generate the trajectory dynamically. Interestingly, under certain cases, NDO is equivalent to FOC and DP respectively.

The core of an NDO model is a multilayer perceptron (MLP) neural network, which generates the control inputs for a non-linear discrete system. The inputs of the MLP are the time-varying states of the MIMO systems at discrete times. The outputs of the MLP, the control parameters, are used by the MIMO to generate new system states. By such a formulation, an NDO model can approximate the time-varying optimal feedback solution.

In our approach, we model the reentry vehicle as a nonlinear MIMO hybrid system⁴. A hybrid system is loosely defined as a system that involves the interaction of discrete event and continuous time dynamics. Hence, to determine the values of system states $x[k+1]$ at next step the continuous form of differential equations of the plant is integrated for the given sampling time, starting from the previous state $x[k]$. The control inputs $u[k]$ during this integration is held constant. Hence, the inputs for the hybrid system model are the current system states and the control parameters, and the outputs are the new system states. Given a task, an ideal trajectory control system will generate a series of control commands to achieve a desired trajectory under various disturbances and vehicle model uncertainties including aerodynamic uncertainties resulting from geometric damage to the vehicle. Conventional control generation approaches suffer from the nonlinearity of the MIMO system and the high-dimensionality of the system state space.

This paper is organized as follows. Section 2 introduces the NDO approach and discusses the implementation issues. Section 3 describes the reentry vehicle dynamics and the path constraints. Section 4 presents the experimental results. Section 5 concludes the paper and discusses the future research direction.

II. Neural Dynamic Optimization

In this section we present the basics of Neural Dynamic Optimization.

A. NDO Formulation

For an optimal control problem, we seek to find a trajectory for a dynamical system that minimize a penalty functional J that maps a Lagrangian function defined over the entire path to a real number. Let a given discrete nonlinear-time-invariant (NLTI) multi-input-multi-output (MIMO) system be defined as:

$$x[k+1] = f(x[k], u[k]), \quad (1)$$

where $x[k] \in R^n$ is the state vector, and $u[k] \in R^m$ is the control vector. The penalty function be defined as:

$$J = \phi(x[N]) + \sum_{k=0}^{N-1} \Gamma(x[k], u[k]), \quad (2)$$

where function $\phi(\cdot)$ penalizes the final state $x[N]$, and the Lagrangian function $\Gamma(\cdot)$ that penalizes the state $x[k]$ and the control inputs $u[k]$ through $k=0$ to $N-1$.

The optimal trajectory design problem can now be stated as: follows

Given an initial state x_0 , find an optimal set of controls $u^*[k]_{k=0,N-1}$ over N steps that minimizes the penalty functional J based on system state history, final system state, and the history of control effort.

Figure 1 illustrates the schematics of a basic NDO-based optimizer, which consists of two components: a representative system model for the given dynamical system; and a neural net model representing the feedback controller. In many applications, the exact system function may be highly non-linear and complex, and hard to obtain in analytic form. Hence we use a representative system model $\tilde{f}(\cdot)$ that approximates the true system dynamics function $f(\cdot)$. The system model $\tilde{f}(\cdot)$ projects the current states to the system's new state based on the current state and the control inputs. The controller $g(\cdot)$ generates a control vector based on the current system state. The controller and the system simulator are coupled so that they can generate a series of states and controls that guides the system in optimal manner to the final desired state $x[N]$.

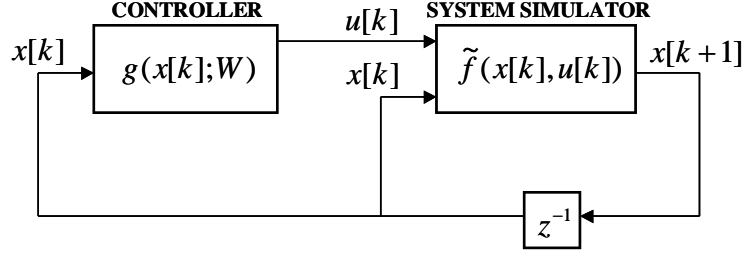


Figure 1. Diagram of an NDO based optimizer.

The challenge of developing an NDO-based controller is to develop a high-performance MIMO controller to handle the complexities arising due to nonlinearity of the plant system. It is expected that optimal trajectories for non-linear system will have a non-linear controller with respect to the system states. We take advantage of the neural networks, structures of massively connected computational units, to handle this complexity. With non-linear activation function, the network has the ability to generalize model optimal controller as a function of system states. The learning capability of a neural network $g(x[k]; W)$ is governed by its structure and the values of its weight matrix, denoted as W . Suppose a full connected three-layer (one input layer, one hidden layer, one output layer) multilayer perceptron (MLP) is selected; the controller designing is reduced to finding an effective means to fine tune W .

In the following part of this section, we will discuss how to train the neural network based on optimal control theory so that its output will fulfill the control goal.

B. Training an NDO model

Consider that the initial system state x_0 lies in a state space described by a probability distribution $P(x_0)$, the NDO based trajectory design solution can be described as:

Minimize cost function J defined in Eqn. (1) with respect to weights W subject to the constraints

$$x[k+1] = f(x[k], u[k]), \quad k=0, \dots, N-1, \quad (3)$$

$$\text{where} \quad x[0] = x_0 \sim P(x_0) \quad (4)$$

$$\text{and} \quad u[k] = g(x[k]; W), \quad k=0, \dots, N-1 \quad (5)$$

Note that both the system equation Eq. (3) and control equations Eq. (5) are treated as constraints. We solve the constrained optimization problem using Euler Lagrange equations. First, we construct the augmented cost function

by adding constraints using Lagrange multipliers $\lambda_x[k]$ and $\lambda_u[k]$. Note that the Lagrange variables are also known as co-states of the adjoint system. The augmented penalty function is given as:

$$\begin{aligned} \bar{J} = & \phi(x[N]) + \sum_{k=0}^{N-1} \Gamma(x[k], u[k]) + \lambda_x^T[0](x_0 - x[0]) \\ & + \sum_{k=0}^{N-1} \lambda_x^T[k+1](f(x[k], u[k]) - x[k+1]) + \sum_{k=0}^{N-1} \lambda_u^T[k](g(x[k]; W) - u[k]) \end{aligned} \quad (6)$$

The solution to the constrained optimization is determined by the saddle point of the augmented penalty function $\bar{J}(x[k], u[k], W, \lambda_x[k], \lambda_u[k])$, which must be minimized with respect to W , $x[k]$, $u[k]$, $\lambda_x[k]$ and $\lambda_u[k]$. Note that for optimal solution, the first variation of the augmented penalty function $\bar{J}(\cdot)$ must be zero. Thus, by differentiating $\bar{J}(x[k], u[k], \lambda_x[k], \lambda_u[k], W)$ with respect to W , $x[k]$, $u[k]$, $\lambda_x[k]$ and $\lambda_u[k]$, and setting the results equal to zero we get the following conditions for the optimality. Note that, the first variation with respect to Lagrange multipliers $\lambda_x[k]$ and $\lambda_u[k]$ returns the constraint equations given by Eqs. (3) and (5) respectively. Whereas the first variation with other parameters yields the following conditions:

Condition 1: First variation with respect to the state variables $x[k]$ is given as:

$$\frac{\partial \bar{J}(x[k], u[k], \lambda_x[k], \lambda_u[k], W)}{\partial x[k]} = 0, \quad k=0, \dots, N-1, \quad (7)$$

which yields the customary *costate* equations for an adjoint discrete system defined as

$$\lambda_x^T[k] = \frac{\partial \Gamma([k], u[k], k)}{\partial x[k]} + \lambda_x^T[k+1] \frac{\partial f([k], u[k], k)}{\partial x[k]} + \lambda_u^T[k] \frac{\partial g(x[k]; W)}{\partial x[k]}, \quad k=0, \dots, N-1. \quad (8)$$

In the above equation we notice the coupling of *costates* λ_x corresponding to system states x , with *costates* λ_u that corresponds to control parameters u . The final boundary conditions for the above adjoint system are determined from the first variation with respect to $x[N]$ as

$$\frac{\partial \bar{J}(x[N], u[N], \lambda_x[N], \lambda_u[N], W)}{\partial x[N]} = 0, \quad (9)$$

which yields
$$\lambda_x^T[N] = \frac{\partial \phi(x[N])}{\partial x[N]}. \quad (10)$$

Condition 2: First variation with respect to the control parameters $u[k]$:

$$\frac{\partial \bar{J}(x[k], u[k], \lambda_x[k], \lambda_u[k], W)}{\partial u[k]} = 0, \quad k=0, \dots, N-1, \quad (11)$$

which yields the other adjoint system with costates λ_u

$$\lambda_u^T[k] = \frac{\partial \Gamma([k], u[k], k)}{\partial u[k]} + \lambda_x^T[k+1] \frac{\partial f([k], u[k], k)}{\partial u[k]}, \quad k=0, \dots, N-1 \quad (12)$$

This adjoint similar is similar to conventional adjoint system in optimal control problems and it is the artifact of the constraints placed on control parameters by the controller function.

Condition 3: First variation with respect to the neural net weights W gives the optimality conditions given as

$$\frac{\partial \bar{J}(x[k], u[k], W, \lambda_x[k], \lambda_u[k])}{\partial W} = 0, \quad k=0, \dots, N-1, \quad (13)$$

which yields

$$\sum_{k=0}^{N-1} \lambda_u^T[k] \frac{\partial g(x[k]; W)^T}{\partial W} = 0, \quad k=0, \dots, N-1. \quad (14)$$

The optimality conditions determines the neural net weight W as function of the history of costates λ_u .

We use a stochastic steepest decent algorithm to find an optimal weight matrix W . Fig. 2 summarizes this process.

Initialize time horizon N , a learning rate μ , an initial guess of W ;
 Define penalty functions $\phi(\cdot)$ and $\Gamma(\cdot)$;

1. Pick $x[0]$ based on Eqn. (3);
2. For $k = 0, \dots, N-1$, compute $x[k+1]$ and $u[k]$ based on Eqn. (2) and (4);
3. Compute $\lambda_x^T[N]$ based on Eqn. (7);
4. For $k = N-1, \dots, 1$, compute $\lambda_x^T[N]$ and $\lambda_u^T[k]$ based on Eqn. (6) and (8) respectively;
5. Update W : $W \leftarrow W - \mu \sum_{k=0}^{N-1} \lambda_u^T[k] \frac{\partial g(x[k]; W)^T}{\partial W}$;
6. Go back to setp 1 or output W if stop criterion is reached.

Fig. 2. Pseudo Code of Stochastic Steepest Descent Algorithm for NDO

C. NDO model for infinite horizon problems

In the previous sections we discussed NDO models with finite time horizon N . The design of an NDO following that scheme does not support solving infinite time horizon problems directly. However, one can solve such problems using such an NDO approach by assigning N a sufficiently large value. If N is sufficiently large, the solution to the finite horizon problem converges to the solution to the corresponding infinite horizon problem. Ref. [2] provides detailed analysis on how to set the initial neural network structure, and Ref.[3] illustrates NDO with an infinite horizon problem.

III. System Model

The following section defines the dynamics of the reentry vehicles.

A. System Functions

The vehicle model used in this paper is a vertical takeoff, horizontal landing, winged-body, and unmanned craft studied at the NASA Marshall Space Flight Center⁵. We are interested in determining a re-entry trajectory that brings the vehicle from the entry point of upper atmosphere to a comparatively low altitude with a relative low velocity. More specifically, under normal conditions, the vehicle enters the atmosphere at an altitude of around 120 km with a speed of approximately 7450 m/s. The intended goal is to determine a feasible trajectory guiding the vehicle to reach to an altitude around 25 km with a speed of approximately 750 m/s subject to any midpoint trajectory feasibility constraints. Table 1 summarizes the symbols used in the paper regarding the vehicle model and the guidance problem.

Table 1. List of symbols and the notations

Symbol	Notation
r	radial distance from the center of the Earth to the vehicle, normalized by R_e
V	Earth-relative velocity, normalized by $\sqrt{g_0 R_e}$
γ	flight path angle (degree)
D	normalized drag in g (acceleration due to gravity)
L	normalized lift in g
C_L	lift coefficient
C_D	drag coefficient
α	angle of attack (degree)
σ	bank angle (degree)
m	mass of the vehicle (104,305 kg)
S	vehicle reference area (391.22 m ²)
R_e	Earth radius (6,378 km)
g_0	acceleration due to gravity at sea level (9.81m/s ²)
ρ	air density, $\bar{\rho}\{1-(25.512r-25.432)\cos[2\pi(129.96-127.56r)]\}$, where $\bar{\rho} = 1.752e^{-R_e(r-1)/6700}$ kg/m ³

In this example, only the normalized longitudinal dynamics of the vehicle are studied. The 2-DOF longitudinal dynamics of the system are modeled using normalized states r , V , and γ . The non-linear longitudinal dynamics of the system is defined by the following system functions:

$$\dot{r} = V \sin \gamma, \quad (15)$$

$$\dot{V} = -D - \sin \gamma / r^2, \quad (16)$$

$$\dot{\gamma} = (V^2 - 1/r) \cos \gamma / (Vr) + (D/V)(C_L / C_D) \cos \sigma, \quad (17)$$

where the drag D , the drag and lift coefficients C_L and C_D are calculated from the following equations:

$$C_L = -0.041065 + 0.016292 \alpha + 0.00026024 \alpha^2, \quad (18)$$

$$C_D = 0.080505 - 0.03026 C_L + 0.86495 C_L^2, \quad (19)$$

$$D = 0.5 \rho V^2 R_e S C_D / m, \quad (20)$$

where angle of attack α is in degrees. For this experiment, an exponential atmosphere was considered that matches quite closely to the 1976 Standard Atmosphere in the altitude regions of interest in this effort. The exponential expression for atmospheric density ρ is represented as

$$\rho = 1.752e^{-\frac{R_e(r-1)}{6700}} \left(\text{kg/m}^3 \right). \quad (21)$$

The first of the three system equations Eq. (15) defines the kinematics, while the other two equations Eqs. (16) and (17) model the acceleration based on aerodynamic forces such as *Lift* and *Drag*. The two control inputs are angle of attack α and bank angle σ , which influence the Drag force and the direction of Lift force acting on the vehicle. The task of the trajectory design is to generate a series of α and σ so that the vehicle can reach the desired state.

B. System Constraints

There are certain constraints that must be considered to design a feasible trajectory. For example, when the vehicle is at a very high altitude with a high speed, a large α is preferred so that the vehicle can use its bottom surface, which is heat resistant, to avoid vehicle damage. The constraints are as follows:

- 1) Heating rate constraint:

$$\sqrt{\rho}(\sqrt{R_e g_0})^3 V^3 \leq 3.305 \times 10^9 \quad (22)$$

- 2) Load factor constraint in body-normal direction:

$$L \cos \alpha + D \sin \alpha \leq 2(g) \quad (23)$$

where L is the dimensionless lift acceleration in g :

$$L = 0.5 \rho V^2 R_e SC_L / m \quad (24)$$

- 3) Dynamic pressure constraint:

$$D \leq \bar{q}_{\max} SC_D / (mg_0) \quad (25)$$

where $\bar{q}_{\max} = 16,280 \text{ N/m}^2$ is the maximum allowable dynamic pressure.

IV. Experimental Results

Based on the system functions and the constraints, we formulate the components of Eqn. (1) as:

$$\begin{aligned} \phi(x[N]) = & Q_r[N](r[N] - r_d)^2 + Q_v[N](V[N] - V_d)^2 + Q_\gamma(\gamma[N] - \gamma_d)^2 \\ & + Q_h(\sqrt{\rho}(\sqrt{R_e g_0})^3 V[N]^3 - 3.305 \times 10^9)^2 \\ & + Q_l(L[N] \cos(\alpha[N - 1]) + D[N] \sin(\alpha[N - 1]) - 2)^2 \\ & + Q_d(D[N] - \bar{q}_{\max} SC_D[N] / (mg_0))^2 \end{aligned} \quad (20)$$

and $\Gamma(x[0], u[0]) = 0$; for $k = 1, \dots, N-1$,

$$\begin{aligned} \Gamma(x[k], u[k]) = & Q_r[k](r[k] - r_d)^2 + Q_v[k](V[k] - V_d)^2 + Q_\gamma(\gamma[k] - \gamma_d)^2 \\ & + Q_\alpha(\alpha[k] - \alpha_d)^2 + Q_\sigma \sigma[k]^2 \\ & + Q_h(\sqrt{\rho}(\sqrt{R_e g_0})^3 V[k]^3 - 3.305 \times 10^9)^2 \\ & + Q_l(L[k] \cos(\alpha[k - 1]) + D[k] \sin(\alpha[k - 1]) - 2)^2 \\ & + Q_d(D[k] - \bar{q}_{\max} SC_D[k] / (mg_0))^2 \end{aligned} \quad (21)$$

subject to

$$14^\circ \leq \alpha \leq 40^\circ,$$

$$0^\circ \leq \sigma \leq 80^\circ,$$

where the final desired normalized $[r_d \ V_d \ \gamma_d]$ have the values $[1.0039197 \ 0.0948 \ 0]^T$ corresponding to an altitude of 120 km, a velocity of 750 m/s, and a zero flight path angle; $Q_r, Q_v, Q_\gamma, Q_\alpha, Q_\sigma, Q_h, Q_l, Q_d$ are the penalty factors with respect to altitude, velocity, flight path angle, angle of attack, bank angle, heat constraint, load factor constraint, and dynamic pressure constraint.

During training, the initial states are generated by uniformly and randomly selecting the altitude, velocity, and flight path angle in the intervals [25, 120] km, [750, 7500] m/s, and [-10, 0], respectively. The time interval for consecutive samples is 5 seconds. The values of $Q_r, Q_v, Q_\gamma, Q_\alpha, Q_\sigma, Q_h, Q_l, Q_d$ are 1, 0.1, 0.1, 0.1, 0.05, 0.1, 0.1, 0.1, respectively. Since this is an infinite horizon problem, a relatively large value, 100, was set for N . The controller MLP has 15 hidden nodes; its learning rate is 1e-7. The experimental results are displayed in Fig. 3-5. This figure shows the vehicle states and the controls, and also the horizontal distance (labeled as *distance*) the vehicle has traveled. This distance can be calculated based on the kinematic relation

$$\dot{s} = V \cos \gamma. \quad (26)$$

Fig. 3 shows the trajectory of the vehicle from the same entry point as used in [4]. The vehicle successfully descends to the desired state within around 1000 seconds. To further evaluate the capability of the NDO, the NDO was tested with some cases that it had not “seen” during training. Fig. 4 shows trajectories of the vehicle with four different initial states. For all the cases, the initial altitude (50 km) and velocity (3000 m/s) are the same; however, the flight path angles are $0^\circ, -2^\circ, 5^\circ$, and -12° , respectively. Since, during the training phase, all the flight path angles are from the interval $[-10^\circ, 0^\circ]$, the third and fourth cases are challenges for the NDO. In Case 3 it takes the vehicle a much longer time to reach the desired state since the initial motion is directed away from the final desired goal. Case 4 is also very challenging as the vehicle is flying toward the desired altitude with an unsustainable flight path, resulting in higher heating due to the higher velocity at more dense lower altitudes. Without effective controls, the vehicle will have reached the desired altitude, but with a much higher velocity than what is desired. Naturally, the vehicle must be directed to fly upward to reduce the speed of the vehicle, and then descend toward the final desired goal. The NDO controller was not trained with such a strategy, as the training is based on the initial state alone. However, Fig. 4 shows that the NDO successfully meets this difficult challenge and the trajectory of Case 4 is on the expected lines for a feasible solution. In summary, the NDO controller successfully generates a feasible path for the vehicle to the desired final state from all four initial conditions.

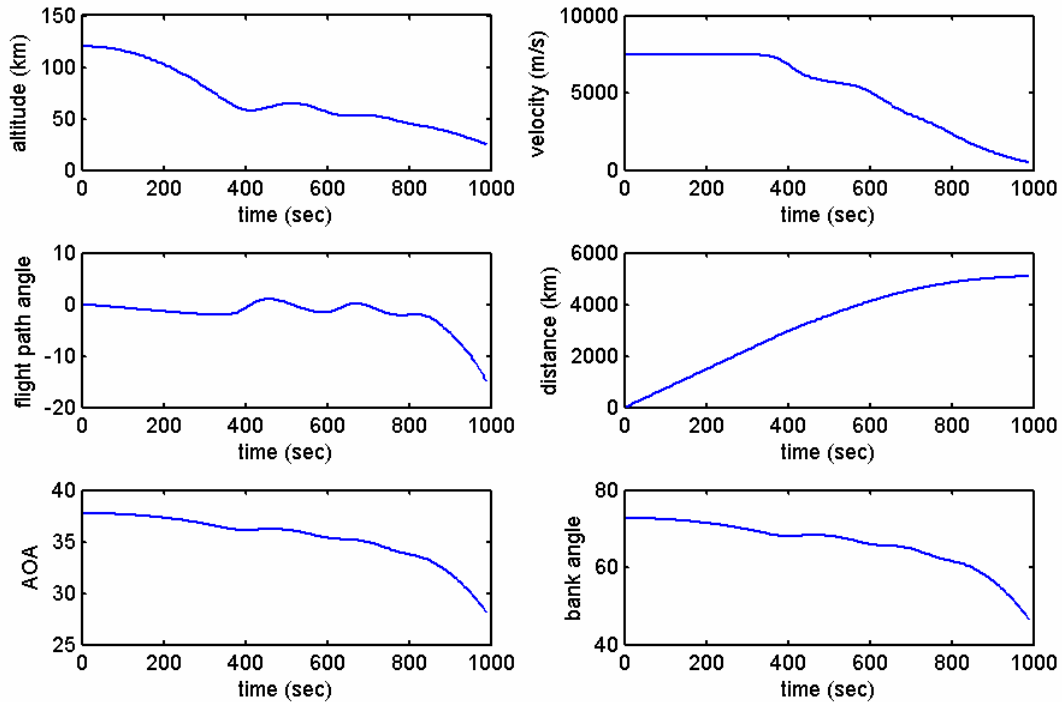


Fig. 3. Trajectory of a Vehicle Entering the Atmosphere at 120 km at the Speed of 7450 m/s, and 0 deg Flight Path Angle.

Finally, to evaluate the robustness of the NDO controller, it is used to design a trajectory for a damaged vehicle. The damaged vehicle was modeled with a 5% reduction in the mass of the vehicle and a loss of 30% of the reference surface area. In this example, the NDO controller faces a difficult challenge as the system dynamics are altered significantly and the NDO controller has no prior knowledge of the change. Notice that the feedback response of the vehicle to the control commands through the measured state is the only indication for the alteration of the system dynamics. Once again, we observe in Fig. 5 that the NDO is capable of generating a feasible path for a significantly damaged vehicle to the desired state.

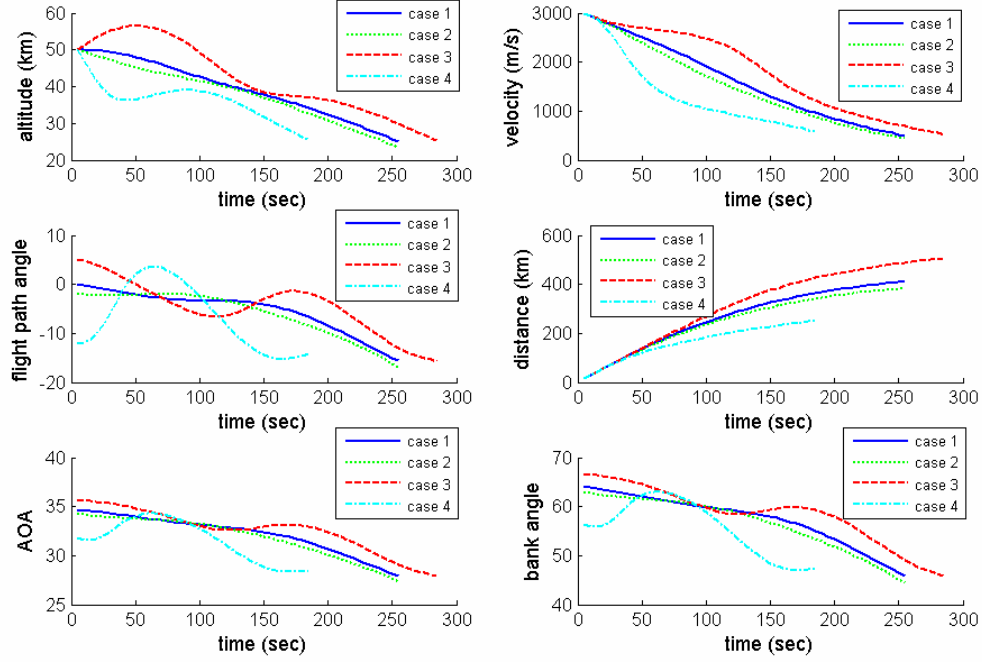


Fig. 4. Trajectories of a Vehicle with Identical Entry Altitude and Velocity, but Different Flight Path Angles

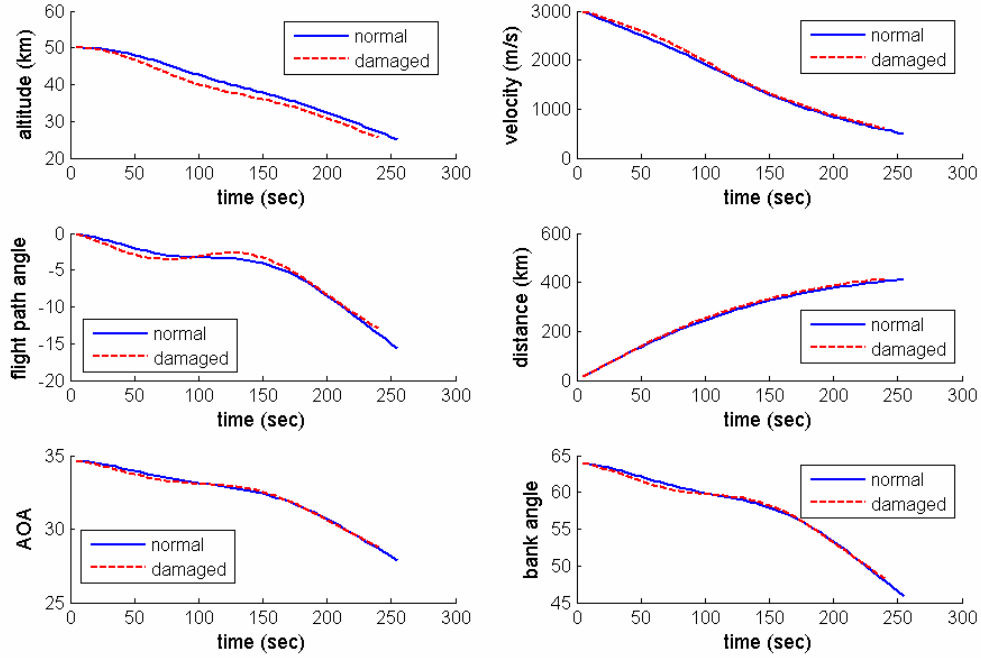


Fig. 5. Trajectories for a Normal Vehicle and a Damaged Vehicle.

V. Conclusion

In this paper, we have explored the theory of Neural Dynamic Optimization (NDO) and applied it to a difficult trajectory design problem for a RLV to design feasible reentry trajectories. Our experimental results show that a properly trained NDO robustly operates for a wide range of input conditions and generates feasible and acceptable trajectories. The results demonstrate that a feasible solution for the cases can be found by the NDO controller even though these have not been included in its training. NDO offers promise as the basis for real-time aerial vehicle trajectory design and reshaping is desired for a class of RLVs in order to achieve more flexibility for autonomous operations. However, drawbacks were also noticed in NDO applications. First, the NDO solution is not a complete dynamic programming solution. This means that local as well as global optima are possible. Second, tuning the weight updating process (by tuning parameters such as learning rate, penalty factors, etc.) is not a trivial task; and requires both detailed analysis of the system function, the penalty function, and also experience in neural network training. Third, the training process is slow since the weight updating follows a sequential mode. In other words, advanced MLP training techniques, which can update weights very fast by analyzing a batch of data points, cannot be easily applied.

The directions we have set for our future work are as follows. First, we need to compare NDO with other control methods [5]; second, we need to study how to adjust the training parameters more systematically; and third, we want to expand the vehicle model to incorporate lateral motion to design a more realistic 3-D trajectory.

Acknowledgments

The research was supported by the AFRL/VACA, WPAFB, Ohio, under contract number FA8650-04-M-3428

References

- ¹ C.-Y. Seong and B. Widrow, "Neural dynamic optimization for control systems-Part I: Background," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 31(4), 482–489 (2001).
- ² C.-Y. Seong and B. Widrow, "Neural dynamic optimization for control systems-Part II: Theory," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 31(4), 490–501 (2001).
- ³ C.-Y. Seong and B. Widrow, "Neural dynamic optimization for control systems-Part III: Applications," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 31(4), 502–513 (2001)

- ⁴ Tomlin, C., Lygeros, J., and Sastry, S. "A game theoretic approach to controller design for hybrid systems," Proceedings of the IEEE, vol. 88, no. 7, pp. 949-970, 2000.
- ⁵ Lu, P. "Entry guidance and trajectory control for reusable launch vehicle," *Journal of Guidance, Control and Dynamics*, 20(1), 143-149 (1997).
- ⁶ Oppenheimer, M. W., Doman, D. B., "Methods for Compensating for Control Allocator and Actuator Interactions," *Journal of Guidance, Control and Dynamics*, 27(5), 922-927 (2004)
- ⁷ Verma, A., Oppenheimer, M. W. , Doman, D. B., "On-Line Adaptive Estimation and Trajectory Reshaping," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 15-18, 2005, San Francisco, CA
- ⁸ Ping Lu, David B. Doman, John D. Schierman, "Adaptive Terminal Guidance for Hypervelocity Impact in Specified Direction," *Journal of Guidance, Control and Dynamics*, 29(2), 269-278 (2006).
- ⁹ D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass., 2003 (second edition).